Automatic Fuzzy Clustering Framework for Image Segmentation

Tao Lei, *Member, IEEE,* Peng Liu, Xiaohong Jia,
Xuande Zhang, Hongying Meng, *Senior Member, IEEE,* and Asoke K. Nandi, *Fellow, IEEE*

***Abstract*—Clustering algorithms by minimizing an object function share a clear drawback that the number of clusters need to be set manually. Although density peak clustering is able to seek the number of clusters, it suffers from memory overflow when it is used for image segmentation because a moderate-size image usually includes a large number of pixels leading to a huge similarity matrix. To address the issue, here we proposed an automatic fuzzy clustering framework (AFCF) for image segmentation. The proposed framework has threefold contributions. Firstly, the idea of superpixel is used for the density peak (DP) algorithm, which efficiently reduces the size of the similarity matrix and thus improves the computational efficiency of the DP algorithm. Secondly, we employ a density balance algorithm to obtain a more robust decision-graph that helps the DP algorithm to achieve fully automatic clustering. Finally, a fuzzy c-means clustering based on prior entropy is used in the framework to improve image segmentation results. Because the spatial neighboring information of both the pixels and membership are considered, the final segmentation result is improved effectively. Experiments show that the proposed framework is not only able to achieve automatic image segmentation, it also provides better segmentation results than state-of-the-art algorithms.**

***Index Terms*—Fuzzy clustering, image segmentation, superpixel, density peak (DP) algorithm**

I. Introduction

CLUSTERING, grouping the objects of a dataset into meaningful subclasses, is one of the most popular research topics, since it is a useful tool for data mining [1], machine learning [2], and computer vision [3]. With the rapid development of intelligent technologies, automated knowledge discovery based on clustering becomes more and more important in these years. Although a large number of clustering algorithms have been successfully used in image segmentation and data classification [4], [5], it is still a challenging topic because it is difficult to achieve automatic clustering and to provide fine results for image segmentation. Image segmentation algorithms based on clustering have two advantages. Firstly, they are able to achieve unsupervised image segmentation without labels. Secondly, they have a better robustness than other image segmentation algorithms such as active contour models [6], graph cuts [7], random walkers [8], region merging [9], etc., since they require fewer parameters. Finally, clustering has a clear advantage on multi-channel image segmentation because it is easy to apply clustering algorithms to high-dimensional data classification. Inevitably, clustering has some disadvantages for image segmentation as well. On the one hand, it is sensitive to noise because the local spatial information of pixels is missed. On the other hand, it takes a long running time when it is used for high-resolution images, as repeated calculations and an iterative optimization are required for the same pixels.

To address the first shortcoming, a simple idea is to incorporate local spatial information into objective functions to improve the robustness of algorithms to noise, such as fuzzy c-means (FCM) clustering algorithm with spatial constraints (FCM_S) [10], FCM_S1/S2 [11], fuzzy local information c-means clustering algorithm (FLICM) [12], neighborhood weighted FCM clustering algorithm (NWFCM) [13], the FLICM based on kernel metric and weighted fuzzy factor (KWFLICM) [14], and deviation-sparse fuzzy c-means with neighbor information constraint (DSFCM_N) [15], etc. Although these improved algorithms can obtain better segmentation of images corrupted by noise, they have two limitations. One is that they need a longer running time than conventional fuzzy c-means clustering algorithm due to the high computational complexity. Moreover, the running time is much worse when these algorithms are used for color image segmentation, for the spatial neighboring information is calculated in each iteration. The other one is that these algorithms employ a fixed neighboring window for each pixel in an image, which leads to a poor segmentation result. For this problem, an instinctive idea is to employ adaptive neighboring information to improve segmentation results. Liu *et al*. [16] improved FCM algorithm by integrating the distance between different regions obtained by mean-shift [17] and the distance of pixels into its objective function. However, as the algorithm employs adaptive neighboring information, its computational complexity is still very high, which limits its practicability in image segmentation.

For the second shortcoming, because the number of gray

This work was supported in part by the National Natural Science Foundation of China under Grant 61871259, Grant 61811530325 (NSFC-RC), Grant 61461025, Grant 61871260, Grant 61672333, Grant 61873155, in part by China Postdoctoral Science Foundation under Grant 2016M602856.

T. Lei is with the School of Electronical and Information Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China, and also with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China. (E-mail: leitao@sust.edu.cn)

P. Liu, X. Jia, and X. Zhang are with the School of Electronical and Information Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China. (E-mail: liupengctu@foxmail.com, jiaxhsust@163.com, love_truth@126.com)

H. Meng is with the Department of Electronic and Computer Engineering, Brunel University London, Uxbridge, Middlesex, UB8 3PH, U.K. (E-mail: hongying.meng@brunel.ac.uk)

A. K. Nandi is with the Department of Electronic and Computer Engineering, Brunel University London, Uxbridge, Middlesex, UB8 3PH, U.K., and also the Key Laboratory of Embedded Systems and Service Computing, College of Electronic and Information Engineering, Tongji University, Shanghai 200092, China. (E-mail: asoke.nandi@brunel.ac.uk)

levels is much smaller than the number of pixels in an image, researchers often perform clustering algorithms on gray levels instead of pixels to avoid the repeated distance computation, which can indeed reduce the execution time of algorithms such as enhanced FCM (EnFCM) [18], fast generalized FCM algorithm (FGFCM) [19], and fast and robust FCM (FRFCM) [20]. These improved FCM algorithms achieve a high computation efficiency by integrating histogram to its objective function. Unfortunately, it is difficult to extend these algorithms to color images because the histogram of color images is more complex than gray images. Moreover, a new problem is how to reduce the computational complexity of algorithms while efficiently improving the utilization of spatial neighboring information. We have mentioned above that the adaptive neighboring information is better than a fixed neighboring window for image segmentation. To reduce the computational complexity while utilizing the adaptive spatial neighboring information, Gu *et al*. [21] employed a superpixel approach to obtain adaptive neighboring information and to reduce the number of clustering samples. They proposed a fuzzy double c-means clustering based on sparse self-representation (FDCM_SSR). Unfortunately, FDCM_SSR still has a higher computational complexity than most of popular algorithms. Inspired by superpixel technology [22] and EnFCM, Lei *et al*. [23] proposed a superpixel-based fast FCM algorithm (SFFCM) for color image segmentation. SFFCM has two advantages. One is that the proposed watershed transform based on multiscale morphological gradient reconstruction (MMGR-WT) is able to provide an excellent superpixel result that is useful for improving the final clustering result. The other one is that the color histogram is integrated into the objective function of FCM to speed up the implementation of the algorithm. Although SFFCM is excellent for color image segmentation, it requires to set manually the number of clusters. In practical image segmentation, it may be impossible to set the number of clusters for each image in an immense image dataset.

To achieve automatic clustering algorithms, researchers tried to estimate the number of clusters using different algorithms such as eigenvector analysis [24], genetic algorithm [25], the particle swarm optimization [26], the robust learning-based schema [27], etc. Although these algorithms are able to find the number of clusters in any unlabeled data set, they are unsuitable for image segmentation since the spatial information is missed and the corresponding segmentation result is coarse. Moreover, none of them is robust for different kinds of data. Density peaks (DP) algorithm proposed by Rodriguez and Laio [28] addresses the problem; it first finds the local density peaks of data, and then computes the minimal distance between a center and other centers that have higher local density than the center, and finally obtains a decision-graph to achieve fast clustering. However, DP algorithm only provides decision-graph without giving the number of clusters. Wang *et al*. [29] proposed a more robust and effective automatic clustering algorithm to overcome the shortcomings of DP algorithm. Though the new algorithm is able to obtain automatically the number of clusters and provides better experimental results, it is still unsuitable for image segmentation since the spatial information of images is missing.

In this paper, we aim to propose an automatic fuzzy clustering framework (AFCF) for image segmentation. The proposed AFCF is inspired by image superpixel, DP algorithm, and prior entropy-based fuzzy clustering. Although the similarity matrix of an image is often huge, which limits the application of DP in image segmentation, we can use a superpixel algorithm to simplify an image to obtain a small similarity matrix that depends on the number of superpixel. Based on the small similarity matrix, we can compute the corresponding decision graph. To obtain automatic clustering algorithms, we need to improve the decision graph to obtain the number of clusters directly without human-computer interaction. Finally, prior entropy is integrated into FCM to improve segmentation results. The proposed AFCF is able to achieve automatic image segmentation with a high precision. Three advantages of the AFCF are presented.

- AFCF is a fully automatic clustering framework for image segmentation, where the number of clusters is not a required parameter compared to existing clustering algorithms.
- AFCF provides accurate number of clusters and achieves better image segmentation than state-of-the-art algorithms because of the utilization of the spatial information of images and prior entropy.
- AFCF has a low memory demand on the experimental environment compared to algorithms connected with DP algorithm because image superpixel addresses the problem of memory overflow.

The rest of this paper is organized as follows. In Section II, we illustrate motivations of this work. In Section III, we propose our methodology and analyze its superiority. The experimental results on synthetic and real images are described in Section IV. Finally, we present our conclusion in Section V.

II. Motivations

Image segmentation results provided by clustering always depend on the number of clusters. Although researchers proposed a lot of adaptive clustering algorithms [30], [31] by estimating the number of clusters, these algorithms have a low robustness and practicability for automatic image segmentation. DP algorithm is able to generate a decision-graph that is helpful for finding the number of clusters, but it has a high computational complexity when it is used in image segmentation. To achieve automatic clustering for image segmentation, two problems need to be overcome. The first is to remove redundant information of images to obtain a small similarity matrix used for DP algorithm. The second is to improve DP algorithm to obtain accurate number of clusters and furtherly achieve image segmentation. We employ superpixel algorithms to simplify the computation of DP algorithm, and then utilize a density balance algorithm to decide the number of clusters. Furthermore, we use prior entropy to improve image segmentation.

A. Parameter-free Clustering

In the popular clustering algorithms, such as k-means, FCM, and spectral clustering, the number of clusters is set manually.

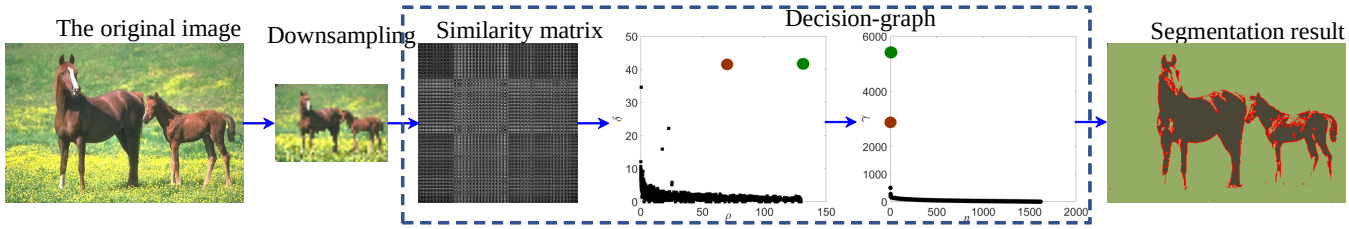


Fig. 1. The image segmentation framework using DP algorithm.

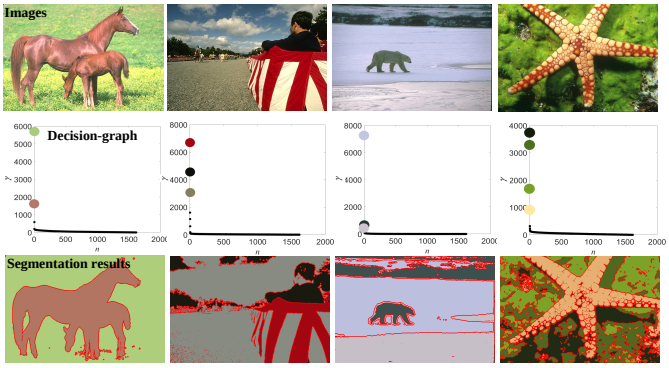


Fig. 2. Segmentation results using DP algorithm. DP algorithm provides a good segmentation result for the first image but poor segmentation results for the last three images.

DP algorithm can automatically recognize potential cluster centers to address the issue by making two basic assumptions. One is that a cluster center often has higher density than its surrounding points. The other is that a cluster center often has a relatively large distance from other cluster centers with high density. Two quantities can be computed for each sample x_i, i.e., the local density ρ_i and the minimal distance δ_i. Both ρ_i and δ_i are obtained from the data. The local density ρ_i of the sample x_i is presented as follows

$$\rho_i = \sum_{j=1, j\neq i}^{N} e^{-\frac{d_{ij}^2}{d_c}}, \tag{1}$$

where N is the number of total samples in a data set, $1 \leqslant i, j \leqslant N$, and d_{ij} denotes the Euclidean distance between x_i and x_j. The d_c is the cutoff distance that is an essential global decay parameter of the weight. The value of d_c is usually around 2% of neighbors [28]. According to (1), ρ_i describes the density intensity of x_i using Gaussian kernel. In general, a large ρ_i is considered as a cluster center while a small ρ_i is considered as noise or outliers in data sets.

δ_i indicates the minimal distance between the sample x_i and any other samples with higher density. The δ_i is defined as

$$\delta_i = \underbrace{min}_{j:\rho_j > \rho_i} (d_{ij}). \tag{2}$$

Note that $\delta_i = max_j(d_{ij})$ is used for the sample with the highest density. The anomalously large value of δ_i is helpful for recognizing the hidden cluster centers. By building a decision-graph with horizontal-axis ρ and vertical-axis δ, we can easily choose the samples of high ρ and relatively high δ as cluster centers. However, it is very difficult to select the appropriate cluster centers for users because there is often a series of continuous sparse points in decision-graphs. To simplify the selection of cluster centers, the DP algorithm designs a new decision scheme by individually computing $\gamma_i = \rho_i \delta_i$ sorted in decreasing order. The new scheme can effectively avoid interference of false centers and easily define the potential centers. After finding the cluster centers, each remaining pixel is allocated to the same cluster as its nearest neighbor of higher density.

Although the decision-graph used for DP is able to provide the potential centers, it is difficult to extend DP algorithm to image segmentation. In the DP algorithm, the computation of ρ and δ depends on a similarity matrix. However, the similarity matrix corresponding to an image of size $P \times Q$, has a large size, i.e., $(P \times Q) \times (P \times Q)$. The matrix always causes memory overflow if P or Q is large. For the problem, the downsampling approach is often used to reduce the size of the matrix to achieve image segmentation based on DP algorithm [32], [33]. Fig. 1 shows the image segmentation framework using DP algorithm. In Fig. 1, one can obtain cluster centers easily because there are two points with large value of ρ and δ in the decision-graph. However, the final segmentation result is coarse. Furthermore, Fig. 2 shows more results generated by DP algorithm.

Figs. 1 and 2 show that the DP algorithm is able to achieve roughly automatic image segmentation. The segmentation result is good when the input image (the first image) is simple, but the results are poor when input images (the last three images) are complex as shown in Fig. 2. To improve these segmentation results, two issues need to be addressed.

- The downsampling operation is a rough way used for the size reduction of a similarity matrix. We need to develop a new algorithm that is not only able to reduce the size of a similarity matrix, but also it can preserve the structuring information of images.
- The decision-graph needs to be improved to obtain automatically the number of clusters.

In this paper, we employ superpixel algorithms instead of the downsampling operation to address the first issue, and then use a density balance algorithm to overcome the second issue. The detailed description will be presented in Section III.

B. Superpixel-based Fast FCM

It is well-known that the iterative optimization is important for clustering algorithms through minimizing an objection

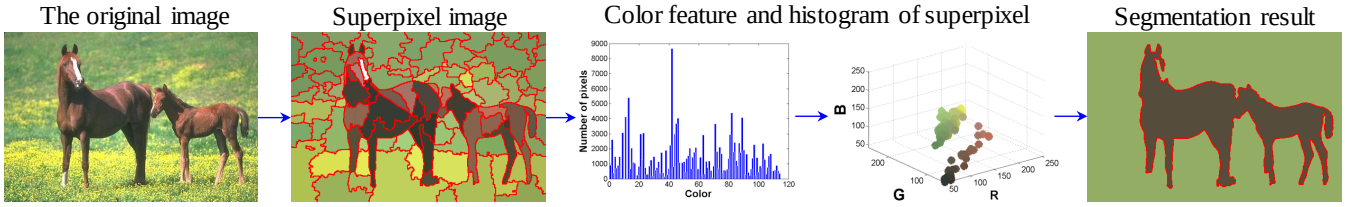


Fig. 3. Image segmentation framework using SFFCM.

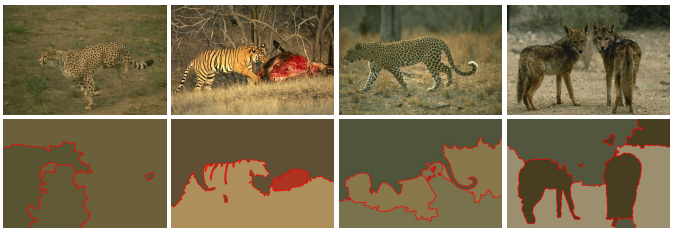


Fig. 4. Error segmentation results using SFFCM (c=3 for each image).

function. Compared with some fast image segmentation algorithms [34], [35], clustering algorithms have a low computational efficiency when they are used to segment an image with high-resolution. Fortunately, superpixel technology [36] plays a key role in improving the execution efficiency of image segmentation algorithms. Superpixel means that an image is divided into a large number of small and independent areas with different sizes and shapes [37]. Based on the superpixel result of an image, one can use a pixel to replace all pixels in a superpixel area to efficiently reduce the number of pixels in an image. Motivated by this, Lei *et al*. proposed SFFCM [23] for color image segmentation. SFFCM addresses two difficulties existing in clustering algorithms for color image segmentation. One is that SFFCM presents an excellent superpixel approach named MMGR-WT, and the superpixel image obtained by MMGR-WT is helpful for improving segmentation effect because the adaptive neighboring information of pixels is integrated into the objective function of clustering algorithms. The other one is that the color histogram is integrated into the objective function to achieve fast clustering due to the fact that the number of different pixels in a color image has been effectively reduced.

The objective function of SFFCM is

$$J=\sum_{l=1}^{N'}\sum_{k=1}^{c}S_l u_{kl}^{m}\left\|(\frac{1}{S_l}\sum_{p\in\partial_l}x_p)-v_k\right\|^2, \tag{3}$$

where ∂_l represents the l-th superpixel area, S_l is the total number of pixels in the superpixel area ∂_l, $1\leqslant l\leqslant N'$. N' is the total number of superpixel area in an image f, and c is the number of clusters. The u_{kl} represents the fuzzy membership of the l-th superpixel area with respect to the k-th cluster, v_k denotes the k-th clustering center, and x_p denotes a pixel in a color image f.

According to (3), the membership partition matrix u_{kl} and the cluster center v_k of SFFCM are given as follows

$$u_{kl}=\frac{\left\|(\frac{1}{S_l}\sum_{p\in\partial_l}x_p)-v_k\right\|^{-2/(m-1)}}{\sum_{j=1}^{c}\left\|(\frac{1}{S_l}\sum_{p\in\partial_l}x_p)-v_j\right\|^{-2/(m-1)}}, \tag{4}$$

$$v_k=\frac{\sum_{l=1}^{N'}u_{kl}^{m}(\sum_{p\in\partial_l}x_p)}{\sum_{l=1}^{N'}S_l u_{kl}^{m}}. \tag{5}$$

It can be seen from (3)-(5) that the computation cost of SFFCM is clearly lower than FCM due to $N'\ll N$. Therefore, the SFFCM achieve fasts and effective color image segmentation. The image segmentation framework based on SFFCM is shown in Fig. 3.

In Fig. 3, the segmentation result is better than the result shown in Fig. 1. The main reason can be attributed to the superpixel image generated by MMR-WT. The result further demonstrates that the advantages of SFFCM. Although SFFCM achieves fast and robust image segmentation, the number of clusters is still an essential parameter, which limits the application of SFFCM. Moreover, the Euclidean distance is used to measure the similarity between different superpixel areas in SFFCM, which causes erroneous segmentation results as shown in Fig. 4.

For the problem shown in Fig. 4, hidden Markov random fields (HMRF) [38], [39] is a popular algorithm used for overcoming the problem. HMRF uses cluster centers and the prior probability of membership to obtain the final membership called posterior probability [40]. Motivated by this, in this work, we employ fuzzy clustering based on prior entropy to achieve image segmentation. The detailed analysis will be presented in Section III.B.

III. Methodology

In Section II, we presented the motivation of this paper. On the one hand, we employ parameter-free clustering to obtain automatically the number of clusters. On the other hand, we employ the image superpixel and fuzzy clustering based on prior entropy to achieve image segmentation. Based on the two ideas, we present the segmentation framework of images using the proposed AFCF as shown in Fig. 5.

In Fig. 5, the proposed AFCF firstly employs a superpixel algorithm to obtain pre-segmentation result. Secondly, DP algorithm is performed on the superpixel image to generate a decision-graph. Because the number of areas in the superpixel image is much smaller than the number of pixels in the original image, a small similarity matrix is obtained resulting in a small

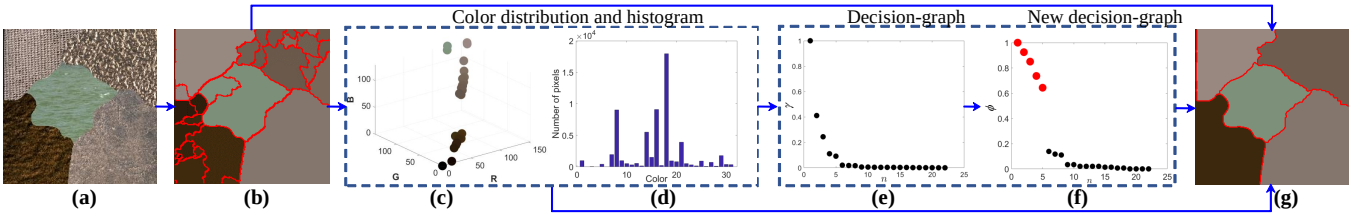


Fig. 5. The proposed image segmentation framework, which is fully automatic for image segmentation. (a) Image. (b) Superpixel image. (c) Color distribution of the superpixel image. (d) Histogram of the superpixel image. (e) Decision-graph obtained by DP algorithm. (f) Improved decision-graph using the density balance algorithm. (g) Segmentation result using FCM based on prior entropy.

memory requirement and a low computational complexity for DP algorithm. After that, the density balance algorithm is used to obtain a more robust decision-graph that directly outputs the number of clusters. By computing the maximal interval of adjacent points, the points in the decision-graph are divided into two groups; the first group of points is considered as cluster centers. Finally, a fuzzy clustering based on prior entropy is used for achieving image segmentation.

A. Decision-graph on Superpixel Images

The DP algorithm achieves semi-automatic clustering because one can choose the number of clusters according to a decision-graph. However, it is difficult to apply DP algorithm to automatic image segmentation because a huge similarity matrix usually causes memory overflow and a high computational cost. For example, a moderate-sized image of size 500×500 generates a huge similarity matrix of size $(500 \times 500)^2$. Such a huge matrix occupies very large memory and requires very high computational cost. Moreover, initial cluster centers obtained by a decision-graph need to be improved for image segmentation because the spatial information of images is missing.

In this section, we firstly employ a superpixel algorithm to reduce the size of the similarity matrix. Because a superpixel algorithm can smooth the texture details and preserve the structuring information of objects, it is helpful for image pre-segmentation to improve the final segmentation effect. Secondly, we use a density balance algorithm and the maximal interval to obtain automatically the number of clusters. In practical applications, because different superpixel algorithms can be selected, we only propose a framework of automatic fuzzy clustering but not a specific algorithm in this paper. Fig. 6 shows superpixel results provided by different superpixel algorithms such as SLIC [41], DBSCAN [42], LSC [43], GMMSP [44], HS [45], and MMGR-WT [23]. Note that each of SLIC, DBSCAN, LSC, and HS requires one parameter, i.e., the number of superpixel area; GMMSP also requires one parameter that is the size of areas; but MMGR-WT needs two parameters that are the initial structuring element denoted by r_1 and the minimal threshold error denoted by η. Table I shows the number of areas in different superpixel images for an original image size of 321×481. In practical applications, η is usually a constant and $\eta = 10^{-4}$ in [23]. We set $\eta = 0.1$ here in order to obtain more superpixel areas for fair comparison.

Table I shows that these superpixel algorithms can efficiently reduce the total number of pixels in an image and thus obtain a small similarity matrix to improve the computational efficiency of DP algorithm. Moreover, a superpixel area integrates both the color features and the spatial structuring features, which is helpful for improving image segmentation results. For instance, the size of the similarity matrix is reduced from $(321 \times 481)^2$ to 200×200 for the SLIC as shown in Fig. 6b.

TABLE I. Comparison of similarity matrix using different superpixel algorithms for an original image size of 321×481.

Algorithms	SLIC	DBSCAN	LSC	GMMSP	HS	MMGR-WT
Number of areas	200	137	182	168	200	147

According to the aforementioned superpixel algorithms and DP algorithm, the local density denoted by ρ_I and the minimal distance denoted by δ_I are presented as follows

$$\rho_I = \sum_{J=1, J\neq I}^{N'} S_J e^{-\frac{D_{IJ}^2}{d_c}}, \tag{6}$$

where $1 \leq I, J \leq N'$, D_{IJ} denotes the Euclidean distance between ∂_I and ∂_J. S_J is the total number of pixels in the J-th superpixel area, d_c is the cutoff distance, and δ_I indicates the minimal distance between the area ∂_I and any other area with higher density. The δ_I is defined as

$$\delta_I = \underbrace{min}_{J:\rho_J > \rho_I} (D_{IJ}), \tag{7}$$

where $\delta_I = \max_J(D_{IJ})$ for the superpixel area with highest density. To speed up the computation, D_{IJ} is defined as

$$D_{IJ} = \left\| \frac{1}{S_I} \sum\nolimits_{p\in\partial_I} x_p - \frac{1}{S_J} \sum\nolimits_{q\in\partial_J} x_q \right\|. \tag{8}$$

It can be seen that D_{IJ} is different from d_{ij}. According to DP algorithm and $\gamma_i = \rho_i \delta_i$, we can obtain the initial decision-graph as shown in Fig. 7. Fig. 7 shows that although one can easily select the number of clusters depending on the decision-graph, it is difficult to obtain automatically the number of clusters by setting a threshold for the decision-graph. In Fig. 7a, the threshold ranges from 0.15 to 0.56, while it ranges from 0.1 to 0.38 for Fig. 7b.

B. Number of Clusters

To achieve fully automatic clustering, we propose a new decision-graph by using a density balance algorithm. We don't need to select the number of clusters and we only compute the maximal interval in the new decision-graph.

(a) (b) (c) (d) (e) (f) (g)

Fig. 6. Superpixel results provided by different superpixel algorithms. (a) Image. (b) SLIC. (c) DBSCAN. (d) LSC. (e) GMMSP. (f) HS. (g) MMGR-WT. The number of superpixel area is 200 for SLIC, DBSCAN, LSC, and HS. For GMMSP, the size of areas is 20×20. For MMGR-WT, $r_1 = 2$, and $\eta = 0.1$.

(a) (b) (c) (d)

Fig. 7. Decision-graphs according to DP algorithm and MMGR-WT. (a) Image 1. (b) Decision-graph of image 1. (c) Image 2. (d) Decision-graph of image 2.

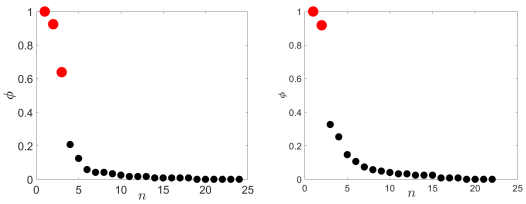


Fig. 8. New decision-graphs of Fig. 7(a) and Fig. 7(c).

The density balance algorithm aims to map the original decision-graph γ_j into a new decision-graph ϕ_j that is superior to γ_j for finding the best number of clusters, where $1 \leq j \leq q$, where q denotes the number of superipixel area. Let a denote the number of intervals in the range [0,1] and κ denote the radius of the neighborhood, where $\chi = \{\chi_1, \chi_2, \ldots \chi_{a+1}\}$ represents the set of data interval, $\chi_1 = 0, \chi_2 = 1/a, \chi_2 = 2/a$, and $\chi_{a+1} = 1$. Generally, a is an empirical value and $a = 1000$. We define that $\xi(\chi_\kappa)$ is the number of γ_j under the constraint condition $\|\chi_a - \gamma_j\| \leq \kappa$, where $1 \leq e \leq a+1$ and γ_j is the normalized result $\gamma_j = (\gamma_j - min(\gamma_j))/(max(\gamma_j) - min(\gamma_j))$, $e, a \in N^+$. $\xi(\chi_e)$ can be computed as follows

$$\xi(\chi_e) = \sum_{j=1}^{N'} \varphi_j, \tag{9}$$

$$\varphi_j = \begin{cases} 1 & \|\chi_e - \gamma_j\| \leq \kappa \\ 0 & otherwise \end{cases}. \tag{10}$$

We define that ϕ_j is the mapping result of γ_j, where ϕ_j can be computed as follows

$$\phi_j = \sum_{\chi_e \leq \gamma_j} \xi(\chi_e). \tag{11}$$

We presented the pseudocode of the algorithm as recorded in Algorithm 1.

According to the Algorithm 1, we compute the new decision-graphs corresponding to Fig. 7(a) and Fig. 7(c). Fig. 8 shows the results.

Algorithm 1: The density balance algorithm

Input: γ_j
Output: ϕ_j
Initialization: set $a = 1000$ and $\eta = 0.1, \chi = \chi_1,\ \chi_2, \ldots \chi_{a+1}$, where $\chi_1 = 0, \chi_2 = 0.001, \chi_3 = 0.002$, and $\chi_{a+1} = 1$.
for $j = 1$ to N', **do**
  for $e = 1$ to $a+1$, **do**
    while $\chi_e \leq \gamma_j$ **do**
      for $b = 1$ to N', **do**
        if $\|\chi_e - \gamma_b\| \leq \kappa$ **then**
          $\varphi_b = 1$
        else
          $\varphi_b = 0$
        end if
        $\xi(\chi_e) = \xi(\chi_e) + \varphi_b$
      end for
      $\phi_j = \phi_j + \xi(\chi_e)$
    end while
  end for
end for

Comparing Fig. 7 and Fig. 8, it is clear that the DB algorithm is useful for finding the best number of clusters. The proposed DB algorithm is able to distinguish efficiently cluster centers and non-cluster centers.

C. Prior Entropy-based Fuzzy Clustering

According to section III. B, we can obtain the number of clusters and the final clustering result using superpixel-based DP algorithm. However, in [28], Euclidean distance is used to measure the similarity between different superpixel areas, which often leads to poor segmentation results for complex images as shown in Fig. 4. We solve this issue using the knowledge that both the covariance analysis and the Markov random field (MRF) are useful for improving high-dimensional data classification [46], [47]. Thus we integrate the covariance analysis and MRF into FCM algorithm, and propose prior entropy-based fuzzy clustering algorithm (PEFC).

Based on superpixel results of images and the number of clusters provided by the Algorithm 1, we firstly propose the

objective function of PEFC as follows

$$J = \sum_{l=1}^{N'} \sum_{k=1}^{c} S_l u_{kl} \Phi(\frac{1}{S_l} \sum\nolimits_{p\in\partial_l} x_p | v_k, \Sigma_k) + \sum_{l=1}^{N'} \sum_{k=1}^{c} S_l u_{kl} \log(\frac{u_{kl}}{\pi_k}) \quad (12)$$

where $\frac{1}{S_l}\sum_{p\in\partial_l} x_p$ is the mean value of the superpixel area ∂_l, and it integrates the spatial information of images. The Σ_k is the covariance matrix that respects to the correlation of different dimensions. The proportion π_k is the prior probability of the superpixel area $\frac{1}{S_l}\sum_{p\in\partial_l} x_p$ belonging to v_k, which satisfies $0 \le \pi_k \le 1$ and $\sum_{k=1}^{c} \pi_k = 1$. The π_k indicates that MRF is integrated into the objective function of PEFC. It is clear that the objective function of PEFC has a more complex structure than FCM, as the PEFC fuses multiple factors that influence image segmentation results.

Furthermore, c is the number of cluster prototypes, which is provided by the Algorithm 1, v_k is the k-th cluster center, u_{kl} represents the membership intensity of the superpixel $\frac{1}{S_l}\sum_{p\in\partial_l} x_p$ with respect to the k-th clustering center v_k, which satisfies $0 \le u_{kl} \le 1$ and $\sum_{k=1}^{c} u_{kl} = 1$, and S_l denotes the number of pixels within the lth superpixel area ∂_l.

In (12), $\Phi(\frac{1}{S_l}\sum_{p\in\partial_l} x_p | v_k, \Sigma_k)$ denotes the multivariate Gaussian distribution, which is defined as follows

$$\Phi(\frac{1}{S_l}\sum_{p\in\partial_l} x_p | v_k, \Sigma_k) = -\log \rho(\frac{1}{S_l}\sum_{p\in\partial_l} x_p | v_k, \Sigma_k), \quad (13)$$

In (13), ρ is the Gaussian density function, which is presented as follows

$$\rho(\frac{1}{S_l}\sum_{p\in\partial_l} x_p | v_k, \Sigma_k) = \frac{\begin{bmatrix} exp(-\frac{1}{2}((\frac{1}{S_l}\sum_{p\in\partial_l} x_p) - v_k)^T \\ \times \sum_k^{-1}((\frac{1}{S_l}\sum_{p\in\partial_l} x_p) - v_k)) \end{bmatrix}}{(2\pi)^{D/2} |\sum_k|^{1/2}}, \quad (14)$$

where D denotes the dimension of image data or the number of image channel, Σ_k is a diagonal matrix of size $D \times D$, and $|\Sigma_k|$ is the determinant of Σ_k. Substituting (14) into (13), we get

$$\begin{aligned} &\Phi(\frac{1}{S_l}\sum_{p\in\partial_l} x_p | v_k, \Sigma_k) \\ &= \frac{1}{2}((\frac{1}{S_l}\sum_{p\in\partial_l} x_p - v_k)^T \Sigma_k^{-1} (\frac{1}{S_l}\sum_{p\in\partial_l} x_p - v_k) \\ &\quad + log\,|\Sigma_k| + Dlog(2\pi)). \end{aligned} \quad (15)$$

According to $\sum_{k=1}^{c} u_{kl} = 1$ and $\sum_{k=1}^{c} \pi_k = 1$, we use the Lagrange multiplier approach to compute the optimal value, and we have

$$\begin{aligned} J = &\sum_{l=1}^{N'} \sum_{k=1}^{c} S_l u_{kl} \Phi(\frac{1}{S_l}\sum\nolimits_{p\in\partial_l} x_p | v_k, \Sigma_k) \\ &+ \sum_{l=1}^{N'} \sum_{k=1}^{c} S_l u_{kl} \log(\frac{u_{kl}}{\pi_k}) \\ &- \lambda_1(\sum\nolimits_{k=1}^{c} u_{kl} - 1) - \lambda_2(\sum\nolimits_{k=1}^{c} \pi_k - 1) \end{aligned} \quad (16)$$

where λ_1 and λ_2 are Lagrange multipliers. Thus we have

$$\frac{\partial J}{\partial u_{kl}} = S_l \Phi(\frac{1}{S_l}\sum_{p\in\partial_l} x_p | v_k, \Sigma_k) + S_l(\log(\frac{u_{kl}}{\pi_k}) + 1) - \lambda_1 = 0.$$

According to $\sum_{k=1}^{c} u_{kl} = 1$, the solution of $\frac{\partial J}{\partial u_{kl}} = 0$ yields

$$u_{kl} = \frac{\pi_k \exp(-\Phi(\frac{1}{S_l}\sum_{p\in\partial_l} x_p | v_k, \Sigma_k))}{\sum_{h=1}^{c} \pi_h \exp(-\Phi(\frac{1}{S_l}\sum_{p\in\partial_l} x_p | v_h, \Sigma_h))}. \quad (17)$$

By computing $\frac{\partial J}{\partial v_k} = 0$, we have

$$\begin{aligned} \frac{\partial J}{\partial v_k} &= \sum_{l=1}^{N'} S_l u_{kl} \left(\frac{\partial \begin{bmatrix} ((\frac{1}{S_l}\sum_{p\in\partial_l} x_p) - v_k)^T \\ \times \Sigma_k^{-1}((\frac{1}{S_l}\sum_{p\in\partial_l} x_p) - v_k) \end{bmatrix}}{\partial v_k} \right) \\ &= \sum_{l=1}^{N'} S_l u_{kl}((\frac{1}{S_l}\sum\nolimits_{p\in\partial_l} x_p) - v_k) \\ &= 0, \end{aligned}$$

and

$$v_k = \frac{\sum_{l=1}^{N'} u_{lk} \sum_{p\in\partial_l} x_p}{\sum_{l=1}^{N'} u_{lk} S_l}. \quad (18)$$

Similarly, we compute $\frac{\partial J}{\partial \Sigma_k} = 0$,

$$\begin{aligned} &\frac{\partial J}{\partial \Sigma_k} \\ &= \sum_{l=1}^{N'} \sum_{k=1}^{c} \frac{1}{2} u_{kl} S_l \left(\frac{\partial \begin{bmatrix} ((\frac{1}{S_l}\Sigma_{p\in\partial_l} x_p) - v_k)^T \\ \times \Sigma_k^{-1}((\frac{1}{S_l}\Sigma_{p\in\partial_l} x_p) - v_k) \end{bmatrix}}{\partial \Sigma_k} + \frac{\partial log\,|\Sigma_k|}{\partial \Sigma_k} \right) \\ &= \sum_{l=1}^{N'} S_l u_{kl}(-((\frac{1}{S_l}\sum_{p\in\partial_l} x_p) - v_k)^T \Sigma_k^{-2}((\frac{1}{S_l}\sum_{p\in\partial_l} x_p) - v_k) + \Sigma_k^{-1}) \\ &= 0 \end{aligned}$$

The solution of $\frac{\partial J}{\partial \Sigma_k} = 0$ yields

$$\Sigma_k = \frac{\sum_{l=1}^{N'} S_l u_{kl}((\frac{1}{S_l}\sum_{p\in\partial_l} x_p) - v_k)^T((\frac{1}{S_l}\sum_{p\in\partial_l} x_p) - v_k)}{\sum_{l=1}^{N'} S_l u_{kl}}, \quad (19)$$

Finally, by computing $\frac{\partial J}{\partial \pi_k} = 0$, we have

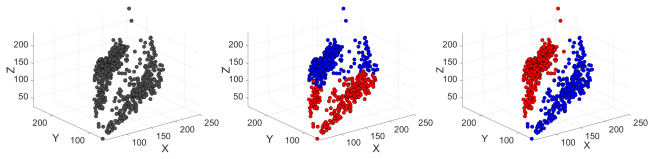
$$\frac{\partial J}{\partial \pi_k} = -\sum_{l=1}^{N'} S_l \frac{u_{kl}}{\pi_k} - \lambda_2 = 0$$

and thus

$$\pi_k = \frac{\sum_{l=1}^{N'} u_{kl} S_l}{\sum_{l=1}^{N'} S_l}. \quad (20)$$

According to (17)-(20), we obtain a membership partition matrix $\mathbf{U} = [u_{kl}]^{c\times N'}$, the cluster center is $\mathbf{V} = [v_k]^{c\times D}$, the covariance matrix $\mathbf{\Sigma} = [\Sigma_k]^{c\times(D\times D)}$, and the proportion $\boldsymbol{\pi} = [\pi_k]^{c\times 1}$.

We can see from (17)-(20) that PEFC integrates the adaptive neighboring information of prior probability distribution, and it considers the distribution characteristic of data. Therefore, it is often used to divide high-dimensional data into different groups. Fig. 9 shows an example where PEFC is used for the



(a) (b) (c)

Fig. 9. Comparison of data classification using FCM and PEFC, respectively. (a) The 3D data. (b) FCM. (c) PEFC.

classification of 3D data.

Based on the analysis of Section III. A.-III. C so far, we propose the detailed steps of the AFCF in the following. The AFCF mainly includes three parts, superpixel pre-processing, improved DP algorithm based on Algorithm 1, and the PEFC.

The detailed steps of AFCF.

Step 1: Compute the superpixel result of the original image, where each area of the superpixel image is denoted by the mean value, $\hat{x}_l = \frac{1}{S_l}\sum_{p\in\partial_l} x_p$. As a result, we get a data set to be classified $\hat{x} = \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_{N'}\}$ and a corresponding color histogram $S = \{S_1, S_2, \cdots, S_{N'}\}$, where S_l denotes the number of total pixels in the superpixel area ∂_l;

Step 2: Implement the DP algorithm on $\hat{x}$ to obtain γ_j;

Step 3: Implement the Algorithm 1 to obtain ϕ_j and the number of clusters;

Step 4: Initialize the variables $\mathbf{U}^{(0)}, \mathbf{V}^{(0)}, \mathbf{\Sigma}^{(0)}$, and $\boldsymbol{\pi}^{(0)}$ using FCM algorithm, where the weighting exponent, the convergence condition, and the maximal number of iterations are 2, 10^{-5}, and 50, respectively;

Step 5: Set the loop counter $t = 0$;

Step 6: Update the variables $\mathbf{U}^{(t)}, \mathbf{V}^{(t)}, \mathbf{\Sigma}^{(t)}$, and $\boldsymbol{\pi}^{(t)}$;

(1) Update the membership matrix $\mathbf{U}^{(t)}$ using (17).

(2) Update the cluster center $\mathbf{V}^{(t)}$ using (18).

(3) Update the covariance $\mathbf{\Sigma}^{(t)}$ using (19).

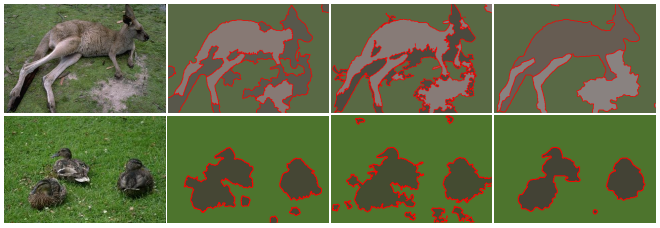
(4) Update the prior probability $\boldsymbol{\pi}^{(t)}$ using (20).

Step 7: If $\max\{\mathbf{U}^{(t)} - \mathbf{U}^{(t+1)}\} < 10^{-5}$ then stop, otherwise, set t=t+1 and go to step 6.

In this paper, three excellent superpixle algorithms, namely SLIC, LSC, and MMGR-WT, are used for the AFCF to obtain three automatic image segmentation algorithms, i.e., SLIC-AFCF, LSC-AFCF, and MMGR-AFCF. Fig. 10 shows the segmentation results generated by the three algorithms. It can be seen that three algorithms obtain the same number of clusters but different segmentation results. MMGR-AFCF obtains a better result than SLIC-AFCF and LSC-AFCF, for MMGR-WT generates a better superpixle result than SLIC and LSC.

IV. Experiments

We conducted experiments on two types of images: synthetic color images with complex texture information (image size is 256×256), as well as real images from the Berkeley segmentation dataset and benchmark (BSDS500) [48] (image size is 481×321 or 321×481). Two synthetic images include three and four different colors and textures, respectively. The BSDS500 includes 300 training images and 200 testing images. There are 4 to 9 ground truth segmentations for each image in BSDS500, and each ground truth is delineated by one human subject. All algorithms and experimental evaluations are performed on a workstation with an Intel(R) Core (TM) CPU, i7-6700, 3.4GHz, and 16GB RAM.



(a) (b) (c) (d)

Fig. 10. Segmentation reuslts using the proposed AFCF. (a) Images. (b) SLIC-AFCF. (c) LSC-AFCF. (d) MMGR-AFCF. The number of superpixel area is 200 for SLIC and LSC. For MMGR-WT, $r_1 = 2$ and $\eta = 10^{-4}$.

To evaluate the effectiveness and efficiency of the proposed AFCF, ten popular image segmentation algorithms based on clustering are considered for comparisons. These are FCM [49], FGFCM [19], HMRF-FCM [40], FLICM [12], NWFCM [13], Liu's algorithm [16], NDFCM [50], FRFCM [20], DSFCM_N [15], and SFFCM [23]. In addition, because the proposed AFCF is an image segmentation framework, three algorithms SLIC-AFCF, LSC-AFCF, and MMGR-AFCF mentioned in Section III. C, are used in our experiments.

A. Parameter Setting

Since both comparative algorithms and the proposed AFCF belong to the class of clustering algorithms through minimizing an objective function. Here three indispensable parameters, i.e., the weighting exponent, the minimal error threshold, and the maximal number of iterations must be set before iterations. In our experiments, the values of these parameters are 2, 10^{-5}, and 50, respectively. The parameter setting of comparative algorithms follows the original paper. As all comparative algorithms require a neighboring window except FCM, HMRF-FCM, FLICM, Liu's algorithm and SFFCM, a window of size 3×3 is used for those algorithms that require a neighboring window for fair comparison. The spatial scale factor and the gray-level scale factor in FGFCM, are $\lambda_s = 3$ and $\lambda_g = 5$, respectively. The NWFCM only refers to the gray-level scale factor, $\lambda_g = 5$. The three parameters, the spatial bandwidth $h_s = 10$, the range bandwidth $h_r = 10$, and the minimum size of final output regions $h_k = 100$ are used for Liu's algorithm. Except three indispensable parameters mentioned above and the number of the cluster prototypes, FCM, HMRF-FCM, FLICM, and DSFCM_N do not require any other parameters. In FRFCM, both the structuring element and the filtering window are a square of size 3×3 for fair comparison. For the SFFCM and the proposed MMGR-AFCF, they share two same parameters used for the MMGR-WT, $r_1 = 2$, and $\eta = 10^{-4}$. For the proposed SLIC-AFCF and LSC-AFCF, the number of superpixel areas is 400 here.

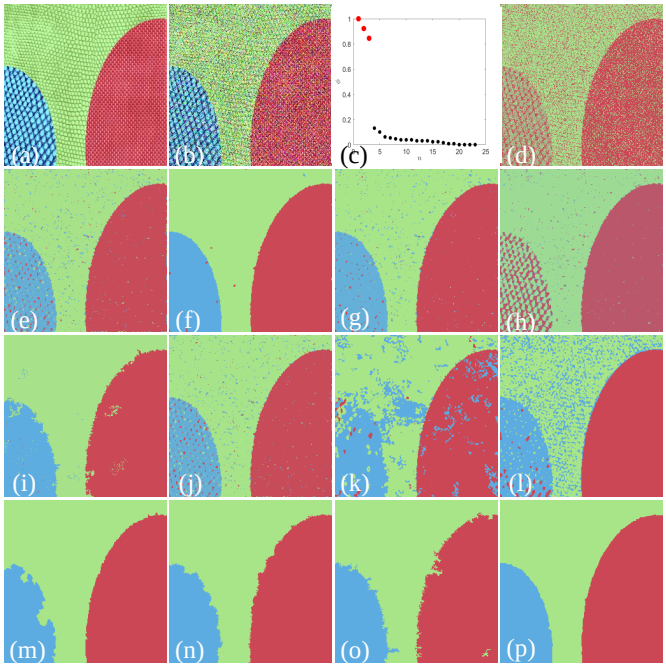


Fig. 11. Comparison of segmentation results on the first symmetric image. (a) The original synthetic image including three areas with different texture and color. (b) The noisy image corrupted by the mixture noise (the mean value is zero and the variance is 0.05 for the Gaussian noise, the density of Salt & Pepper is 0.1). (c) The decision-graph. (d) FCM. (e) FGFCM. (f) HMRF-FCM. (g) FLICM. (h) NWFCM. (i) Liu's algorithm. (j) NDFCM. (k) FRFCM. (l) DSFCM_N. (m) SFFCM. (n) SLIC-AFCF. (o) LSC-AFCF. (p) MMGR-AFCF.

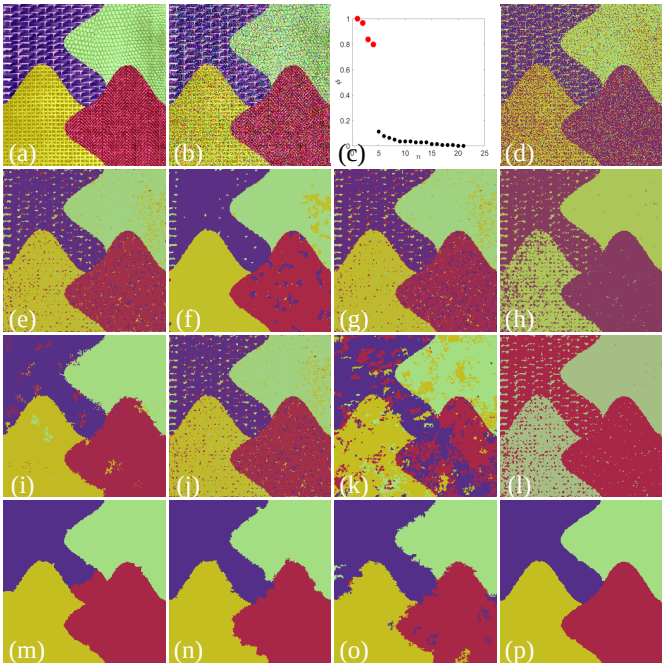


Fig. 12. Comparison of segmentation results on the second symmetric image. (a) The original synthetic image including four areas with different texture and color. (b) The noisy image corrupted by the mixture noise (the mean value is zero and the variance is 0.05 for the Gaussian noise, the density of Salt & Pepper is 0.1). (c) The decision-graph. (d) FCM. (e) FGFCM. (f) HMRF-FCM. (g) FLICM. (h) NWFCM. (i) Liu's algorithm. (j) NDFCM. (k) FRFCM. (l) DSFCM_N. (m) SFFCM. (n) SLIC-AFCF. (o) LSC-AFCF. (p) MMGR-AFCF.

B. Results on Synthetic Images

As this paper proposes a fully automatic fuzzy clustering framework for image segmentation, we demonstrate that the proposed framework is able to provide accurate number of clusters and achieve better image segmentation than comparative algorithms. Two synthetic images are considered as testing images, where we use the texture information of the Colored Brodatz Texture database (http://multibandtexture.recherche.usherbrooke.ca/colored%20_brodatz.html) to generate complex texture images. Furthermore, these texture images are corrupted by different kinds of noise. Two kinds of different noises - Gaussian noise and Salt & Pepper noise are added to two testing images. The final segmentation results are shown in Figs. 11 and 12.

Note that all comparative algorithms require the number of clusters except the proposed AFCF since it is fully automatic for image segmentation. Figs. 11(c) to 12(c) show the final decision-graph and it is clear that the number of clusters c is 3 for Fig. 11 and c is 4 for Fig. 12, which demonstrates that the proposed AFCF is effective for finding the best value of c. For fair comparison, $c = 3$ and $c = 4$ are used for all comparative algorithms on Fig. 11 (a) and Fig. 11 (b), respectively.

Fig. 11 (d) shows that FCM provides a poor segmentation result since the spatial information of images is missing. Although FRFCM and DSFCM_N integrate spatial information of images into their objective function, lots of pixels are classified falsely as shown in Figs. 11 (k and l). FRFCM obtains a poor result, since the multivariate morphological reconstruction is unsuitable for images with complex texture. DSFCM_N employs the sparse representation to improve image segmentation results, but it is not effective for images corrupted by the mixture noise. Figs. 11 (e, g, h, and j) provide good segmentation results because the neighboring information employed by FGFCM, FLICM, NWFCM, and NDFCM is effective for improving segmentation results. Furthermore, Liu's algorithm and HRMF-FCM provide better results, which show that they are robust for images corrupted by noise. AFCF has a strong capability of noise suppression, and MMGR-AFCF obtains the best result that looks similar to the result provided by HMRF-FCM.

Fig. 12 (a) has a more complex shape and texture than Fig. 11 (a). In Figs. 12 (d, e, and l), FCM, FGFCM, and DSFCM_N generate poor segmentation results because they obtain wrong cluster centers, which indicates that the three algorithms have a weak capability of pixel classification for complex images corrupted by noise. Figs. 12 (f, g, h, i, and j) offters similar results, which means that HMRF-FCM, FLICM, NWFCM, NDFCM, and Liu's algorithm have a limited capability for improving segmentation effect on complex images. Both SFFCM and AFCF obtain better segmentation results as shown in Figs. 12 (m, n, o, and p), which shows that the superpixel technology and the prior entropy are effective for improving segmentation results on complex images. Similar to Fig. 11

TABLE II. Scores (S%) of different algorithms on the first synthetic image corrupted by noise (c=3), where SP represents Salt & Pepper noise and G represents Gaussian noise. The best values are highlighted.

Algorithms	SP 10%	SP 20%	SP 30%	G 5%	G 10%	G 15%	SP 10% +G 5%	SP 20% +G 10%	SP 30% +G 15%	Mean Value	RMSE
FCM	84.24	71.43	66.91	73.50	67.76	64.52	69.57	61.84	57.03	68.53	7.72
FGFCM	98.88	96.52	89.40	97.92	93.98	89.18	96.08	83.78	74.71	91.16	7.89
HMRF-FCM	93.69	90.58	88.13	99.84	95.30	89.87	99.80	96.46	83.10	92.97	5.57
FLICM	99.07	94.77	84.05	99.14	96.90	92.85	97.71	85.99	78.68	92.13	7.44
NWFCM	94.77	90.14	82.11	86.56	82.35	81.03	84.48	79.06	74.63	83.90	6.00
Liu's algorithm	99.68	98.43	97.75	99.35	97.65	94.68	97.76	80.17	68.40	92.65	10.91
NDFCM	98.90	96.56	89.32	98.06	94.31	89.53	96.23	84.01	74.94	91.32	7.84
FRFCM	95.75	86.13	85.09	84.40	78.55	76.53	84.13	70.45	64.08	80.57	9.38
DSFCM_N	99.88	86.47	85.84	89.80	81.89	80.24	87.08	78.68	73.99	84.87	7.46
SFFCM	99.29	96.49	93.25	97.80	97.70	95.92	97.78	90.83	90.36	95.49	3.24
SLIC-AFCF	99.48	98.94	98.92	99.21	98.81	98.46	99.16	98.39	87.36	97.64	3.87
LSC-AFCF	99.67	99.41	98.42	98.75	98.19	91.39	97.51	93.83	75.91	94.79	7.60
AMR-AFCF	99.67	99.58	99.39	99.66	99.38	99.24	99.58	99.38	99.23	**99.46**	**0.17**

TABLE III. Segmentation accuracies (SA%) of different algorithms on the second synthetic image corrupted by noise (c=4), where SP represents Salt & Pepper noise and G represents Gaussian noise. The best values are highlighted.

Algorithms	SP 10%	SP 20%	SP 30%	G 5%	G 10%	G 15%	SP 10% +G 5%	SP 20% +G 10%	SP 30% +G 15%	Mean Value	RMSE
FCM	64.99	46.69	37.37	42.10	32.54	29.14	35.68	33.81	24.25	38.51	11.93
FGFCM	91.77	82.53	69.84	86.05	74.13	65.71	78.95	59.96	48.89	73.09	13.53
HMRF-FCM	99.61	98.89	64.96	97.03	82.01	66.47	90.07	52.51	39.35	76.78	21.97
FLICM	85.00	67.99	48.77	88.02	73.07	53.96	74.92	40.57	25.49	61.98	21.10
NWFCM	95.61	44.34	38.69	50.50	33.25	32.30	39.82	31.73	30.87	44.12	20.39
Liu's algorithm	98.97	94.30	86.82	97.59	83.53	68.43	91.34	48.43	29.93	77.70	24.11
NDFCM	91.73	82.27	69.62	86.50	74.62	66.30	79.17	60.33	48.86	73.27	13.50
FRFCM	94.49	85.11	53.89	59.31	43.68	36.99	57.80	40.07	37.32	56.52	20.80
DSFCM_N	97.57	96.31	39.09	68.01	48.83	38.37	32.93	30.95	29.48	53.50	27.28
SFFCM	98.79	97.94	90.58	98.14	89.74	87.60	98.59	84.57	71.33	90.80	9.07
SLIC-AFCF	98.39	96.18	96.78	98.52	96.68	95.90	95.03	94.35	84.4	95.14	4.25
LSC-AFCF	98.44	97.83	95.8	95.80	93.54	82.60	95.23	86.21	70.88	90.70	9.15
AMR-AFCF	99.16	98.81	98.37	99.06	98.90	98.50	98.63	97.57	96.96	**98.44**	**0.73**

(p), the MMGR-AFCF obtains the best segmentation result as shown in Fig. 12 (p).

To evaluate the performance of different algorithms on noisy images, two performance indices are employed here. The first is the quantitative index score (S) that is the degree of equality between pixel sets A_k and the ground truth C_k [20], The second is the optimal segmentation accuracy (SA) that is the sum of the number of the correctly classified pixels divided by the sum of the number of total pixels. They are defined as

$$S = \sum_{k=1}^{c} \frac{A_k \bigcap C_k}{A_k \bigcup C_k}, \tag{21}$$

$$SA = \sum_{k=1}^{c} \frac{A_k \bigcap C_k}{\sum_{j=1}^{c} C_j}, \tag{22}$$

where A_k is the set of pixels belonging to the kth class found by an algorithm while C_k is the set of pixels belonging to the class in the ground truth. All algorithms are performed on the two synthetic images corrupted by different kinds and levels of noise. The experimental results are shown in Tables II-III. As the algorithms used in experiments show different performance for images corrupted by different kinds of noise, we further presented the mean value and the root mean square error (RMSE) of S and SA in Tables II and III.

In Tables II and III, FCM obtains low mean value and high RMSE of S and SA because it is sensitive to both Salt & Pepper noise and Gaussian noise. NWFCM, FRFCM, and DSFCM_N show better performance than FCM, but they are poor compared to other comparative algorithms, especially when the noise level is high. The DSFCM_N is robust for Salt & Pepper noise, but it is sensitive to Gaussian noise, and thus it is difficult to obtain a good segmentation result using the DSFCM_N for images corrupted by the mixture noise.

FGFCM, HMRF-FCM, NDFCM, Liu's algorithm, and FLICM obtain similar average performance for the first synthetic image as shown in Table II, but the FLICM shows worse performance than other four algorithms for the second synthetic image as shown in Table III. SFFCM shows excellent performance for two synthetic images. We can see that SFFCM is insensitive to both Salt & Pepper noise and Gaussian noise. However, it is sensitive to Gaussian noise when the noise level is high, as shown in Table III. The proposed AFCF

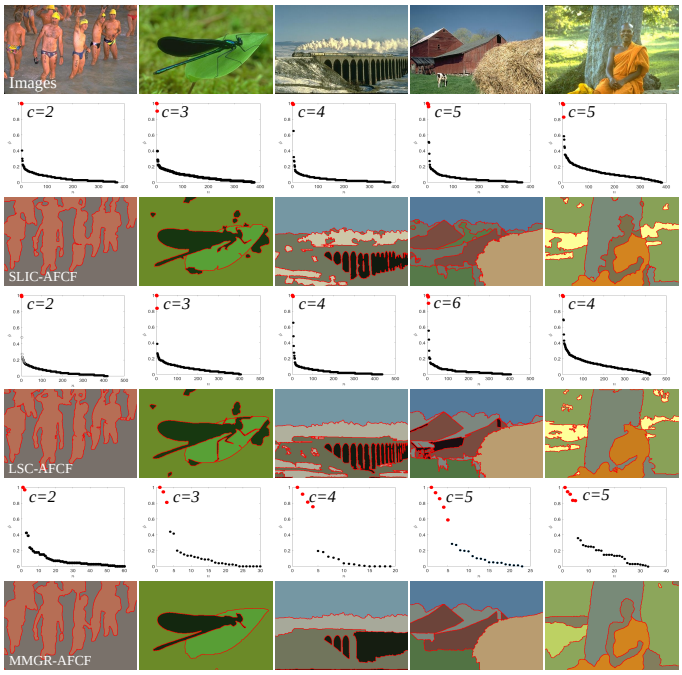


Fig. 13. Segmentation results using the proposed AFCF. The first row denotes original images. The second, fourth, and sixth rows denote decision-graphs obtained by SLIC-AFCF, LSC-AFCF, and MMGR-AFCF, respectively. The third, fifth, and seventh rows are segmentation results generated by the three proposed algorithms.

TABLE IV. Average performance on five images shown in Fig. 13. The best values are highlighted.

Algorithms	c	PRI↑	CV↑	VI↓	GCE↓	BDE↓
SLIC-AFCF	c=2	0.66	0.58	2.06	**0.08**	16.76
	c=3	0.81	0.64	1.82	0.13	9.98
	c=4	0.81	0.60	1.85	0.17	9.83
	c=5	0.81	0.54	1.98	0.25	9.02
	c=6	0.80	0.50	2.07	0.28	9.69
	Adaptive c	**0.87**	**0.69**	**1.61**	0.16	**5.91**
LSC-AFCF	c=2	0.66	0.59	0.96	**0.05**	18.99
	c=3	0.81	0.64	1.79	0.13	9.85
	c=4	0.83	0.62	1.78	0.17	10.41
	c=5	0.83	0.58	1.89	0.22	9.57
	c=6	0.82	0.51	2.07	0.29	9.41
	Adaptive c	**0.85**	**0.65**	**1.66**	0.15	**7.14**
MMGR-AFCF	c=2	0.70	0.59	2.01	**0.08**	12.89
	c=3	0.83	0.64	1.75	0.13	8.70
	c=4	0.82	0.60	1.74	0.17	9.57
	c=5	0.83	0.61	1.66	0.17	8.33
	c=6	0.82	0.56	1.83	0.22	9.00
	Adaptive c	**0.90**	**0.74**	**1.32**	0.10	**6.04**

shows better performance than comparative algorithms except SFFCM. Especially, MMGR-AFCF obtains the largest mean value and the smallest RMSE of S and SA on two synthetic images, which demonstrates that MMGR-AFCF is robust for different kinds of image corrupted by noise.

C. Results on Real Images

The previous experiments demonstrate that the proposed AFCF is useful for automatic image segmentation, and it is robust for images corrupted by different kinds of noise. To demonstrate further the superiority of AFCF on real image segmentation, we conducted experiments on BSDS500. The parameters values in all algorithms are the same as that in Section IV. B except FRFCM and SFFCM. The related parameters of FRFCM and SFFCM follow the original papers [20] and [23]. In addition, the CIE-Lab color space is used for all algorithms for fair comparison.

To evaluate the performance of different algorithms for real image segmentation, we consider the probabilistic rand index (PRI), the covering (CV), the variation of information (VI), the global consistency error (GCE), and the boundary displacement error (BDE) [51], as the performance metrics. Both PRI and CV are usually used for the evaluation of the pixelwise classification task, where PRI is the similarity of labels and CV is the overlap of regions between two clustering results. VI is also used for the purpose of clustering comparison, and it is the distance of average conditional entropy between two clustering results. Additionally, GCE and BDE are often used for the evaluation of image segmentation, where GCE computes the global error to which two segmentations are mutually consistent and BDE measures the average displacement error of boundary pixels between two segmentations. In general, a good segmentation corresponds to high value of PRI and CV, but corresponds to low values of VI, GCE, and BDE.

Firstly, we demonstrate that the proposed AFCF is able to obtain an accurate number of clusters for real images. Because it is difficult to give an accurate number of clusters for each image in BSDS500, we choose five simple images as shown in Fig. 13. According to the proposed AFCF that includes SLIC-AFCF, LSC-AFCF, and MMGR-AFCF, the corresponding decision-graphs and segmentation results are obtained as shown in Fig. 13. We can see that SLIC-AFCF, LSC-AFCF, and MMGR-AFCF obtain the approximate number of clusters, which means that AFCF obtains and provides similar number of clusters for an image independent of the selected superpixel algorithm.

In Fig. 13, SLIC-AFCF and MMGR-AFCF obtain the same number of cluster for each image, but LSC-AFCF obtains different results for images “97010” and “376020”. In practical applications, because three superpixel algorithms lead to different pre-segmentation results, it is impossible to obtain same number of clusters for each image in BSDS500 using three algorithms. However, Fig. 13 shows that three suprepixel algorithms can obtain an approaximate number of clusters. Because the MMGR-WT has been able to provide better superpixel results, we consider the number of clusters provided by MMGR-AFCF for comparative algorithms in the following experiments.

To demonstrate further that AFCF is able to generate the best number of clusters, Table V shows the performance of the proposed AFCF on real image segmentation. Because our purpose is to demonstrate that the proposed AFCF can obtain the best number of clusters, we present the average values of

Fig. 14. Comparison of segmentation results on images from the BSDS500 using different algorithms.

Fig. 15. Comparison of segmentation results on images from the BSDS500 using different algorithms.

PRI, CV, VI, GCE, and BDE on five images shown in Fig. 13. The proposed AFCF can automatically obtain the value of c, which means that the AFCF obtains a value of c adaptively. However, the conventional clustering algorithms use a fixed value of c for each image. In Table IV, the fixed value of c and the adaptive value of c are used for AFCF, respectively. It is clear that the adaptive value of c is superior to the fixed value of c, which demonstrate that the proposed AFCF is effective for obtaining an accurate number of clusters.

To demonstrate further that the proposed AFCF is effective for most images in the BSDS500, we performed all algorithms on each image in BSDS500. For fair comparison, the same value of c obtained by the proposed MMGR-AFCF is used for all algorithms. Experimental results are shown in Figs. 14 and 15. Moreover, Table V shows the performance comparison of different algorithms on BSDS500.

In Figs. 14 and 15, we can see that FCM, FGFCM, HMRF-FCM, FLICM, NWFCM, KWFLICM, NDFCM, FRFCM, and DSFCM_N generate segmentation results including a great number of isolated regions. Therefore, it is difficult to obtain fine contours for segmentation results. The main reason is that these algorithms employ fixed-size windows to obtain spatial neighboring information. The Liu's algorithm, SFFCM, and the proposed AFCF obtain better segmentation results due to the employment of superpixel algorithms, which means that the adaptive spatial information is useful for improving segmentations. However, because MMGR-WT always generates better superpixel areas than SLIC and LSC, MMGR-AFCF provides better segmentation results than SLIC-AFCF and LSC-AFCF. Compared to SFFCM that employs Sobel operators, MMGR-AFCF employs structured forests (SE) [52]

TABLE V. The performance comparison of different algorithms on BSDS500 dataset. The best values are highlighted.

Algorithms	PRI↑	CV↑	VI↓	GCE↓	BDE↓
FCM	0.70	0.41	2.87	0.37	14.01
FGFCM	0.69	0.40	2.92	0.38	14.29
HMRF-FCM	0.72	0.44	2.59	0.33	14.22
FLICM	0.71	0.43	2.73	0.35	13.47
NWFCM	0.71	0.42	2.79	0.36	13.70
Liu's algorithm	0.74	0.47	2.47	0.30	**12.95**
NDFCM	0.69	0.40	2.93	0.38	14.38
FRFCM	0.71	0.44	2.67	0.34	13.52
DSFCM_N	0.70	0.40	2.92	0.38	14.48
SFFCM	0.73	0.50	2.18	0.25	14.13
SLIC-AFCF	0.71	0.47	2.43	0.28	14.55
LSC-AFCF	0.73	0.47	2.44	0.29	13.98
MMGR-AFCF	**0.76**	**0.54**	**2.05**	**0.22**	**12.95**

TABLE VI. Comparison of execution time (in seconds) of different algorithms. The best values are highlighted.

Algorithms	FCM	FRFCM	DSFCM_N	SFFCM	SLIC -AFCF	LSC -AFCF	MMGR -AFCF
Time	1.15	1.87	19.15	**0.74**	2.44	2.88	1.05

to generate a gradient image to be used for MMGR. Therefore, MMGR-AFCF provides better superpixel results than SFFCM. Furthermore, as the former employs prior entropy to obtain the final result, it has clear advantages than the later for image segmentation.

In Table V, FCM, FGFCM, FLICM, NWFCM, NDFCM, and DSFCM_N obtain similar values of PRI, CV, VI, GCE and BDE, which shows that these algorithms have similar performance on real image segmentation. Similarly, HMRF-FCM has a similar performance with FRFCM. Liu's algorithm clearly outperforms other algorithms due to the employment of superpixel images generated by mean-shift algorithm. SFFCM obtains better CV, VI, GCE but worse PRI and BDE than Liu's algorithm. The proposed SLIC-AFCF and LSC-AFCF show similar performance with Liu's algorithm. However, they are superior to Liu's algorithm since the proposed AFCF is fully automatic and it has lower computational complexity than Liu's algorithm. The proposed MMGR-AFCF provides the best results in each of the five performance metrics, which demonstrates that MMGR-AFCF is able to obtain excellent segmentation results for real images.

D. Execution Time

Execution time is important because it evaluates the practicability of a segmentation algorithm. We computed the execution time of different algorithms on all images in the BSDS500, and Table VI shows the comparison of average execution time on 500 images. In all comparative algorithms, it is known that the FLICM, HMRF-FCM, NWFCM, and Liu's algorithm are quite time-consuming [16]. Moreover, FRFCM has a lower computational complexity than FGFCM and NDFCM [50]. Therefore, we only present the execution time of FCM, FRFCM, DSFCM_N, SFFCM, and the proposed SLIC-AFCF, LSC-AFCM, MMGR-AFCF in Table VI.

In Table VI, we can see that SFFCM is the fastest due to the utilization of superpixel and color histogram. DSFCM_N needs a long execution time because the neighboring information is repeatedly computed in each iteration. The proposed SLIC-FCM and LSC-FCM require a similar execution time that is longer than MMGR-AFCF, for both SLIC and LSC has a higher computational complexity than MMGR-WT. Clearly, the proposed MMGR-AFCF is not only faster than all comparative algorithms except SFFCM, but also it is fully automatic for image segmentation.

V. Conclusion

In this paper, we have studied automatic image segmentation algorithms using fuzzy clustering. We proposed an automatic fuzzy clustering framework for image segmentation by integrating superpixel algorithms, density peak clustering, and prior entropy. The proposed AFCF addresses two difficulties that exists in popular algorithms. One is that the proposed AFCF is fully automatic for image segmentation since the number of clusters is obtained automatically. The other one is that the proposed AFCF provides better image segmentation results than popular clustering algorithms due to the employment of superpixel algorithms and the prior entropy. The proposed AFCF was used to segment synthetic and real images. The experimental results demonstrate that the proposed AFCF is able to obtain accurate number of clusters. Moreover, the AFCF is superior to state-of-the-art clustering algorithms because it provides the best segmentation results.

However, we only use the average color in a superpixel area as the features of the superpixel area, which is a drawback for image segmentation. In the future, we will explore feature learning algorithm to achieve better automatic image segmentation algorithms, and we will consider convolutional neural networks (CNN) to extract image features.

References

[1] A. Fahad, N. Alshatri and Z. Tari, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis," *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 267-279, Sept. 2014.

[2] M. T. Law, R. Urtasun and R. S. Zemel, "Deep spectral clustering learning," in *Proc. Int. Conf. Machine Learn.*, Sydney, Australia, 2017, pp: 1985-1994.

[3] G. Nebehay and R. Pflugfelder, "Clustering of static-adaptive correspondences for deformable object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 2784-2791.

[4] G. Dong and M. Xie, "Color clustering and learning for image segmentation based on neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 925-936, Jul. 2005.

[5] N. Kumar, P. Uppala and K. Duddu, "Hyperspectral tissue image segmentation using semi-supervised NMF and hierarchical clustering," *IEEE Trans. Med. Imag.*, 2018. Doi: 10.1109/TMI.2018.2883301.

[6] K. Zhang, L. Zhang, K. M. Lam and D. Zhang, "A level set approach to image segmentation with intensity inhomogeneity," *IEEE Trans. Cybern*, vol. 46, no. 2, pp. 546-557, Feb. 2016.

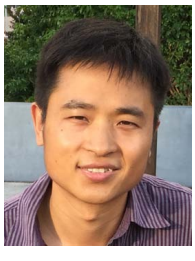
[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888-905. Aug. 2000.

[8] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452-1458. Nov. 2004.

[9] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1452-1458. Sept. 2004.

[10] M. N. Ahmed, S. M. Yamany, N. A. Mohamed, A.A. Farag and T. Moriarty, "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data," *IEEE Trans. Med. Imag.*, vol. 21, no. 3, pp. 193-199, Mar. 2002.

[11] S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1907-1916, Aug. 2004.
[12] S. Krinidis and V. Chatzis, "A robust fuzzy local information c-means clustering algorithm," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1328-1337, May 2010.
[13] Z. Zhao, L. Cheng and G. Cheng, "Neighbourhood weighted fuzzy cmeans clustering algorithm for image segmentation," *IET Image Process.*, vol. 8, no. 3, pp. 150-161, Mar. 2014.
[14] M. Gong, Y. Liang, J. Shi, W. Ma and J. Ma, "Fuzzy c-means clustering with local information and kernel metric for image segmentation," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 573-584, Feb. 2013.
[15] Y. Zhang, X. Bai, R. Fan and Z. Wang, "Deviation-sparse fuzzy c-means with neighbor information constraint," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 1, pp. 185-199, Jan. 2019.
[16] G. Liu, Y. Zhang and A. Wang, "Incorporating adaptive local information into fuzzy clustering for image segmentation," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3990-4000, Nov. 2015.
[17] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[18] L. Szilágyi, Z. Benyó, S. Szilágyii and H. Adam, "MR brain image segmentation using an enhanced fuzzy c-means algorithm," in *Proc. 25th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Cancun, MX, 2003, pp. 17-21.
[19] W. Cai, S. Chen, and D. Zhang, "Fast and robust fuzzy c-means clustering algorithms incorporating local information for image segmentation," *Pattern Recognit.*, vol. 40, no. 3, pp. 825-838, Mar. 2007.
[20] T. Lei, X. Jia, Y. Zhang, L. He, H. Meng and A. K. Nandi, "Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3027-3041, Oct. 2018.
[21] J. Gu, L. Jiao, S. Yang and F. Liu, "Fuzzy double c-means clustering based on sparse self-representation," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 612-626, Apr. 2018.
[22] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson and K. Siddiqi, "Turbopixels: fast superpixels using geometric flows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009.
[23] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng and A. K. Nandi, "Superpixel-based fast fuzzy c-means clustering for color image segmentation," *IEEE Trans. Fuzzy Syst.*, 2019. Doi: 10.1109/TFUZZ.2018.2889018.
[24] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Proc. Syst.*, Vancouver, CA, 2005, pp. 1601-1608.
[25] L. Y. Tseng and S. B. Yang, "A genetic approach to the automatic clustering problem," *Pattern Recognit.*, vol. 34, no. 2, pp. 415-424, Feb. 2001.
[26] S. Das, A. Abraham and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp: 218-237, Jan. 2008.
[27] M. S. Yang and Y. Nataliani, "Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters," *Pattern Recognit.*, vol. 71, pp. 45-59, Nov. 2017.
[28] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp: 1492-1496. Jun. 2014.
[29] G. Wang and Q. Song, "Automatic clustering via outward statistical testing on density metrics," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp: 1971-1985, Aug. 2016.
[30] Y. Zhu, K. M. Ting and M. J. Carman, "Density-ratio based clustering for discovering clusters with varying densities," *Pattern Recognit.*, vol. 60, pp. 983-997, Dec. 2016.
[31] R. J. Kuo and F. E. Zulvia, "Automatic clustering using an improved artificial bee colony optimization for customer segmentation," *Knowl. Inf. Syst.*, vol. 57, no. 2, pp. 331-357, Nov. 2018.
[32] J. Hou, H. Gao and X. Li, "DSets-DBSCAN: a parameter-free clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp: 3182-3193, Jul. 2016.
[33] J. Hou and W. Liu, "A parameter-independent clustering framework," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1825-1832, Aug. 2017.
[34] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision.*, vol. 59, no. 2, pp. 167-181, Sept. 2004.
[35] H. Lombaert, Y. Sun, L. Grady and C. Xu, "A multilevel banded graph cuts method for fast image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Beijing, CN, 2005, pp. 259-265.
[36] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed and G. Jones, "Superpixel lattices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1-8.
[37] J. Shen, Y. Du, W. Wang and X. Li, "Lazy random walks for superpixel segmentation," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1451-1462, Apr. 2014.
[38] F. Forbes and N. Peyrard, "Hidden Markov random field model selection criteria based on mean field-like approximations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1089-1101, Sept. 2003.
[39] A. Quattoni, S. Wang, L. P. Morency, M. Collins and T. Darrell, "Hidden conditional random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1848-1852, Oct. 2007.
[40] S. P. Chatzis and T. A. Varvarigou, "A fuzzy clustering approach toward hidden Markov random field models for enhanced spatially constrained image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1351-1361, Oct. 2008.
[41] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274-2282, Nov. 2012.
[42] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang and L. Shao, "Real-time superpixel segmentation by DBSCAN clustering algorithm," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5933-5942, Dec. 2016.
[43] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 1356-1363.
[44] Z. Ban, J. Liu and L. Cao, "Superpixel segmentation using gaussian mixture model," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 4105-4117, Aug. 2018.
[45] X. Wei, Q. Yang, Y. Gong and N. Ahuja, "Superpixel hierarchy," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4838-4849, Oct. 2018.
[46] M. Gong, L. Su, M. Jia and W. Chen, "Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 1, pp. 98-109, Feb. 2014.
[47] J. Hu and Y. Wen, "Adaptive fuzzy clustering algorithm with local information and Markov random field for image segmentation," in *Proc. Int. Conf. Neural Inf. Process.*, Springer, CH, 2018, pp. 170-180.
[48] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898-916, May. 2011.
[49] J. C. Bezdek, R. Ehrlich and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, no. 2-3, pp. 191-203, May. 1984.
[50] F. Guo, X. Wang and J. Shen, "Adaptive fuzzy c-means algorithm based on local noise detecting for image segmentation," *IET Image Process.*, vol. 10, no. 4, pp. 272-279, Apr. 2016.
[51] X. Wang, Y. Tang, S. Masnou and L. Chen, "A global/local affinity graph for image segmentation," *IEEE Trans. Image Process.*, vol. 24, no. 4, pp. 1399-1411, Apr. 2015.
[52] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1558-1570, Aug. 2015.



Tao Lei (M'17) received the Ph.D degree in Information and Communication Engineering from Northwestern Polytechnical University, Xi'an, China, in 2011. From 2012 to 2014, he was a Postdoctoral Research Fellow with the School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China. From 2015 to 2016, he was a Visiting Scholar with the Quantum Computation and Intelligent Systems group at University of Technology Sydney, Sydney, Australia. He is currently a Professor with the School of Electrical and Information Engineering, Shaanxi University of Science and Technology. His current research interests include image processing, pattern recognition, and machine learning.

Peng Liu received the B.S. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, and the M.S. degree from Chengdu University of Information and Technology, Chengdu, China. He is currently pursuing the Ph.D. degree at the School of Electronical and Information Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China. His current research interests include computer vision and pattern recognition.

Xiaohong Jia received the M.S. degree in Signal and Information Processing in Lanzhou Jiaotong University, Lanzhou, China, in 2017. He is currently pursuing the Ph.D. degree at the School of Electronical and Information Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China. His current research interests include image processing and pattern recognition.

Xuande Zhang received the B.S. degrees from Ningxia University, Yinchuan, China, in 2000, He received his M.S. and PH.D. degree from Xidian University, Xi'an, China, in 2006 and 2013, respectively. From 2009 to 2010, he was a Visiting Scholar with the Department of Computing at The Hong Kong Polytechnic University. He is currently a professor with the School of Electrical and Information Engineering, Shanxi University of Science and Technology. His researching interest includes image processing and pattern recognition.

Hongying Meng (M'10-SM'17) received his Ph.D. degree in Communication and Electronic Systems from Xi'an Jiaotong University, Xi'an China, in 1998. He is currently a senior lecturer at the Department of Electronic and Computer Engineering, Brunel University London, U.K. He is also a member of Institute of Environment, Health and Societies, Human Centred Design Institute (HCDI), and Wireless Networks and Communications Research Center at Brunel. He is a Fellow of The Higher Education Academy (HEA) and a member of Engineering Professors Council in UK.

Dr Meng's current research interests include digital signal processing, affective computing, machine learning, human computer interaction, computer vision, and embedded systems with over 90 publications in these areas. Especially, his audio based and video based emotion recognition systems have won the International Audio/Visual Emotion Challenges AVEC2011 and AVEC2013 prizes respectively. He is a senior member of the IEEE.

Asoke K. Nandi (F'11) received the degree of Ph.D. in Physics from the University of Cambridge (Trinity College), Cambridge (UK). He held academic positions in several universities, including Oxford (UK), Imperial College London (UK), Strathclyde (UK), and Liverpool (UK) as well as Finland Distinguished Professorship in Jyvaskyla (Finland). In 2013 he moved to Brunel University London (UK), to become the Chair and Head of Electronic and Computer Engineering. Professor Nandi is a Distinguished Visiting Professor at Tongji University (China) and an Adjunct Professor at University of Calgary (Canada).

In 1983 Professor Nandi co-discovered the three fundamental particles known as W^+, W^- and Z^0 (by the UA1 team at CERN), providing the evidence for the unification of the electromagnetic and weak forces, for which the Nobel Committee for Physics in 1984 awarded the prize to his two team leaders for their decisive contributions. His current research interests lie in the areas of signal processing and machine learning, with applications to communications, gene expression data, functional magnetic resonance data, and biomedical data. He has made many fundamental theoretical and algorithmic contributions to many aspects of signal processing and machine learning. He has much expertise in "Big Data", dealing with heterogeneous data, and extracting information from multiple datasets obtained in different laboratories and different times.

He has authored over 580 technical publications, including 230 journal papers as well as four books, entitled Automatic Modulation Classification: Principles, Algorithms and Applications (Wiley, 2015), Integrative Cluster Analysis in Bioinformatics (Wiley, 2015), Blind Estimation Using Higher-Order Statistics (Springer, 1999), and Automatic Modulation Recognition of Communications Signals (Springer, 1996). Recently he published in Blood, International Journal of Neural Systems, BMC Bioinformatics, IEEE TWC, NeuroImage, PLOS ONE, Royal Society Interface, Mechanical systems and Signal Processing, and Molecular Cancer. The h-index of his publications is 67 (Google Scholar) and ERDOS number is 2.

Professor Nandi is a Fellow of the Royal Academy of Engineering (UK) and also a Fellow of seven other institutions, including the IEEE and the IET. Among the many awards he received are the Institute of Electrical and Electronics Engineers (USA) Heinrich Hertz Award in 2012, the Glory of Bengal Award for his outstanding achievements in scientific research in 2010, the Water Arbitration Prize of the Institution of Mechanical Engineers (UK) in 1999, and the Mountbatten Premium, Division Award of the Electronics and Communications Division, of the Institution of Electrical Engineers (UK) in 1998. Professor Nandi is an IEEE EMBS Distinguished Lecturer (2018-19).